**Programming in C Quiz Set 2:**

1. **void main()**

**{**

**char \*p=”hello…””world”;**

**printf (p);**

**}**

1. **void main()**

**{ printf (“\nab”);**

**printf (“\bsi”);**

**printf (“\rha”);**

**}**

1. **void main()**

**{**

**int a, b=4;**

**a=printf (“DDP=%d”, b) + 3;**

**printf (“%d”, a);**

**}**

1. **How many tokens are there in given statements?**

**printf (“%d%d”, &a, b);**

1. **main()**

**{ unsigned a=-1;**

**printf (“%u”, ++a);**

**}**

1. **main ()**

**{ unsigned int a=0xffff;**

**~a;**

**printf (“%x”, a);**

**}**

1. **main ()**

**{ 200;**

**printf (“%d”, 200);**

**}**

1. **main ()**

**{ int x=32767;**

**printf (“%d”, x<<1);**

**}**

1. **main ()**

**{ int i=10, j=11, k, l;**

**k =i+++j;**

**l =i+++ ++j;**

**printf (“ %d %d %d %d, i, j, k, l );**

**}**

1. **main ()**

**{**

**int k=1;**

**printf (“%d==1 is” “%s”, k, k ==1? “True” : “False”);**

**}**

1. **main ()**

**{ int i=1, j=3;**

**switch (i)**

**{**

**case 1:**

**printf (“outside1\n”);**

**switch (j)**

**{**

**case 3:**

**printf (“inside\n”);**

**break;**

**}**

**case 2:**

**printf(“ outside2\n”);**

**}**

**}**

1. **main ()**

**{ int i=1;**

**for (; ;)**

**{ printf(“%d”, i);**

**if(i=5)**

**break;**

**}**

**}**

1. **main ()**

**{ int i=3;**

**for (;i++=0;)**

**printf (“%d”, i);**

**}**

1. **main ()**

**{**

**int a=0, b=20;**

**char x=1, y=10;**

**if (y, x, b, a)**

**printf (“C language”);**

**}**

1. **main ()**

**{int \*a;**

**float \*b;**

**char \*c;**

**printf (“%d%d%d”, sizeof (a), sizeof (b), sizeof(c));**

**}**

1. **main()**

**{ int a[5]={1, 2, 3, 4, 5}; //assume base address is 2001**

**printf(“%p %p”, a, &a);**

**printf (“%p %p”, a+1, &a+1);**

**}**

1. **main ()**

**{ int a=20;**

**int \*p=&a;**

**printf (“%d %d”,++\*p, \*p++);**

**}**

1. **main ()**

**{ int a=20;**

**int \*p=&a;**

**printf (“%d %d”,\*p++, ++\*p);**

**}**

1. **main ()**

**{ int a=20;**

**const int \*p=&a;**

**\*p=40;**

**printf (“%d ”,\*p);**

**}**

1. **main ()**

**{ int a=20, b=30;**

**int \*const p=&a;**

**\*p=50;**

**printf (“%d”, \*p);**

**p=&b;**

**\*p=10;**

**printf (“%d”,\*p);**

**}**

1. **main ()**

**{ int a=20, b=30;**

**const int \*const p=&a;**

**\*p=50;**

**printf (“%d”, \*p);**

**p=&b;**

**\*p=10;**

**printf (“%d ”,\*p);**

**}**

1. **main ()**

**{ int a[]={20, 40, 80};**

**a[0, 1, 2]=10;**

**printf(“%d %d %d”, a[0], a[1], a[2]);**

**}**

1. **main ()**

**{ int a[]={20, 40, 80, 100};**

**a[ 1+ 2]=10;**

**printf(“%d %d %d %d”, a[0], a[1], a[2], a[3]);**

**}**

1. **main ()**

**{ int a[][]={20, 40, 80, 100};**

**printf(“%d %d %d %d”, a[0][0], a[0][1], a[1][0], a[1][1]);**

**}**

1. **main ()**

**{ int a[]={0, 1, 2, 3, 4};**

**int \*p[]= {a, a+1, a+2, a+3, a+4};**

**int \*\*ptr=p;**

**printf (“%d %p %p %p %p %p”, \*\*ptr, &ptr, \*ptr, \*p, p, a);**

**}**

1. **main ()**

**{ int a[2][2][2]={ 1, 2, 3, 4, 5, 6, 7, 8};**

**printf (“%p %p %p\n”, a, a[0], a[0][0]);**

**printf (“%p %p %p\n”, a, a[1], a[1][1]);**

**printf(“%d %d \n”, a[0][0][0], a[1][1][1]);**

**}**

1. **How do you declare and initialize and call a pointer to a function?**
2. **What is an array of function pointers ? How does it work?**
3. **main ()**

**{ int a=2;**

**printf (“%d”, sizeof ( printf(“ Hello DDP”));**

**printf(“%d”, sizeof (a+=2));**

**printf(“%d”, a);**

**}**

1. **fun1()**

**{ return printf(“ in fun1\n”);**

**}**

**fun2 ()**

**{ return printf(“in fun2\n”);**

**}**

**main()**

**{ printf(“%d”, fun1()+fun2());**

**}**

**PROBLEM:**

1. Write a C function for addition of two numbers without using Arithmetic Operators (Maximum in Linear time)
2. Write a C function to check whether the input number is even or odd without using modulus operator in constant time.
3. Write a C function to set a particular bit position equal to one of a given integer number. Display the updated result.
4. A sorted array is partially rotated, you have to find minimum element of this partial rotated array.
5. Calculate value of angle created between hour and minute hands at a particular time in a clock.